Jerry Zhu

347-481-1012 jerry.zhu.153@gmail.com linkedin.com/in/jerzhu github.com/Xpost2000

Technical Skills

Languages: C++, C, C#, GLSL, HLSL, Lua, Basic x86 Assembly and SSE intrinsics
Skills: REST APIs, Test Driven Development, CI/CD, Agile Software Development
Technologies: OpenGL, Direct3D11, SDL2, Win32 API
Tools: Visual Studio, Git, RenderDoc, GDB, Valgrind, Bash
Game Engines: Unity, Godot

Experience

Explore (SWE & PM) Intern — C#, ASP.NET, Azure **May 2024 - August 2024**
Microsoft - Xbox Platform & Experiences Redmond, WA

- Implemented player test account distribution within an ASP.NET backend service for game testing enforcing single point of presence to facilitate online/networked multiplayer game testing setup.
- Developed features such as allocated device export/import to ease integration into existing game tester workflows involving Game Development Kit (GDK) tools such as Xbox One Manager.
- Designed unit tests, and end to end (E2E) integration tests to increase code coverage and ensure feature robustness through CI/CD pipelines.
- Gathered feature requirements, wrote, and iterated on design specifications to augment a game testing service with multiplayer testing support for first party game studios such as Turn10.

Education

Stony Brook University **August 2022 - May 2026**
Bachelor of Science, Computer Science Stony Brook, NY

- **Cumulative GPA**: 3.65
- **Coursework**: Operating Systems, Linear Algebra, Computational Geometry, Data Structures & Algorithms

Projects

2D SHMUP - Soulstorm — C++, Lua, Custom Engine

- Designed a command-buffer based renderer abstraction to facilitate supporting three different graphics API targets including Direct3D11, OpenGL 3.3, and software renderer backends.
- Implemented a memory-efficient replay system through input-journaling and a deterministic game loop through the use of fixed time step and custom random number generation.
- Reduced graphics API draw calls by 50% with an in-engine load-time sprite atlas builder to allow for automatic batching of sprite rendering operations.
- Accelerated and parallelized game entity and particle update loops through a multithreaded work queue system.
- Integrated Lua scripting with a coroutine based task scheduler for designing stage encounters and patterns.

3D Dungeon Crawler - Soul Walker — C#, Unity

- Designed a configurable actor controller utilizing callbacks for events used for the player and 4 enemy types.
- Developed a raycast based movement system with support for sloped surfaces and custom surface behaviors.
- Implemented a pickup system for item types such as keys, weapons, potions and an in-game 3D inventory display.

2D Tactical RPG - Legends — C, Custom Engine

- Developed a software renderer with support for 3 blend modes, sprite rotation and scaling optimized with tiled multithreaded jobs and SSE improving performance by 200%.
- Implemented a multithreaded post-processing pipeline for a software renderer with custom shaders.
- Implemented an initiative turn-based combat system through a queue with undo-able actions.
- Eliminated runtime memory allocations with a double ended stack allocator and object pools backed by a fixed 16MB buffer.
- Designed a backward-compatible save system with delta-compression to reduce file size and memory usage.

Custom Language Transpiler - CrankLang — C++

- Designed a language supporting user-defined types, and multiple file modules with a recursive descent parser.
- Compiles programs from a type-checked abstract syntax tree (AST) into C++ code.
- Implemented a constant-folding optimization pass which simplifies the AST by determining and precalculating all constant expressions before output.